M9-L1 Problem 2

Recall the von Mises stress prediction problem from the module 6 homework. In this problem, you will compute the R^2 score for a few model predictions for a single shape in this dataset. You will also plot the predicted-vs-actual stress for each model.

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import r2_score

float32 = np.float32
```

In [2]:
```
xs = np.array([0.        , 1.        , 1.        , 0.84258664, 0.8588491 , 0.86      ,
ys = np.array([1.        , 1.        , 0.        , 0.1574128 , 0.19719249, 0.24159406,
gt = np.array([0.08839864, 0.08831116, 0.19658579, 0.12091552, 0.22751924, 0.2393936 ,
model1 = np.array([ 1.11028813e-01,  1.22077152e-01,  1.65442377e-01,  1.55162349e-01,
model2 = np.array([0.14395204, 0.14395204, 0.14395204, 0.14395204, 0.14395204, 0.14395
model3 = np.array([ 2.81959862e-01,  4.73277241e-01,  2.35924363e-01,  6.43386170e-02,
```

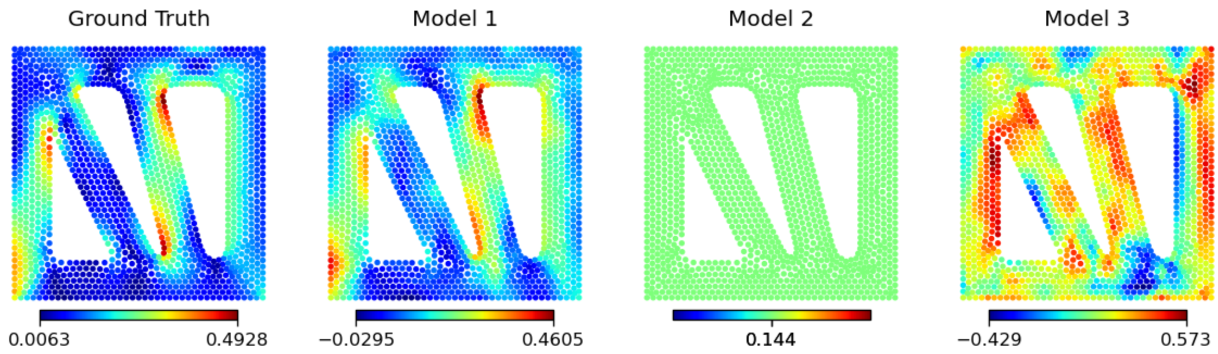
Visualizing data

Run the following cell to load the data and visualize the 3 model predictions.

- `gt` is the ground truth von Mises stress vector
- `model1` is the vector of stress predictions for model 1
- `model2` is the vector of stress predictions for model 2
- `model3` is the vector of stress predictions for model 3

In [3]:
```
def plot_shape(x, y, stress, lims=None):

    if lims is None:
        lims = [min(stress),max(stress)]

    plt.scatter(x,y,s=5,c=stress,cmap="jet",vmin=lims[0],vmax=lims[1])
    plt.colorbar(orientation="horizontal", shrink=.75, pad=0,ticks=lims)
    plt.axis("off")
    plt.axis("equal")

def plot_all(x, y, gt, model1, model2, model3):
    plt.figure(figsize=[12,3.2], dpi=120)
    plt.subplot(141)
    plot_shape(x, y, gt)
    plt.title("Ground Truth")

    plt.subplot(142)
    plot_shape(x, y, model1)
    plt.title("Model 1")

    plt.subplot(143)
    plot_shape(x, y, model2)
    plt.title("Model 2")
```

```
    plt.subplot(144)
    plot_shape(x, y, model3)
    plt.title("Model 3")

    plt.show()

plot_all(xs, ys, gt, model1, model2, model3)
```



Computing R^2

Calculate the R^2 value for each model and print the results.

In [6]:
```
# YOUR CODE GOES HERE

#For model 1
S_tot_1 = 0
S_res_1 = 0

n = len(model1)
y_m = np.mean(model1)
for i in range(len(model1)):
    S_tot_1 += (gt[i]-y_m)**2
    S_res_1 += (gt[i] - model1[i])**2
R_sq_1 = 1- (S_res_1/S_tot_1)
print(f'R_squared for Model 1: {R_sq_1}')

#For model 2
S_tot_2 = 0
S_res_2 = 0

n = len(model2)
y_m = np.mean(model2)
for i in range(len(model2)):
    S_tot_2 += (gt[i]-y_m)**2
    S_res_2 += (gt[i] - model2[i])**2
R_sq_2 = 1- (S_res_2/S_tot_2)
print(f'R_squared for Model 2: {R_sq_2}')

#For model 3
S_tot_3 = 0
S_res_3 = 0

n = len(model3)
y_m = np.mean(model3)
for i in range(len(model3)):
```

```
    S_tot_3 += (gt[i]-y_m)**2
    S_res_3 += (gt[i] - model3[i])**2
R_sq_3 = 1- (S_res_3/S_tot_3)
print(f'R_squared for Model 3: {R_sq_3}')
```

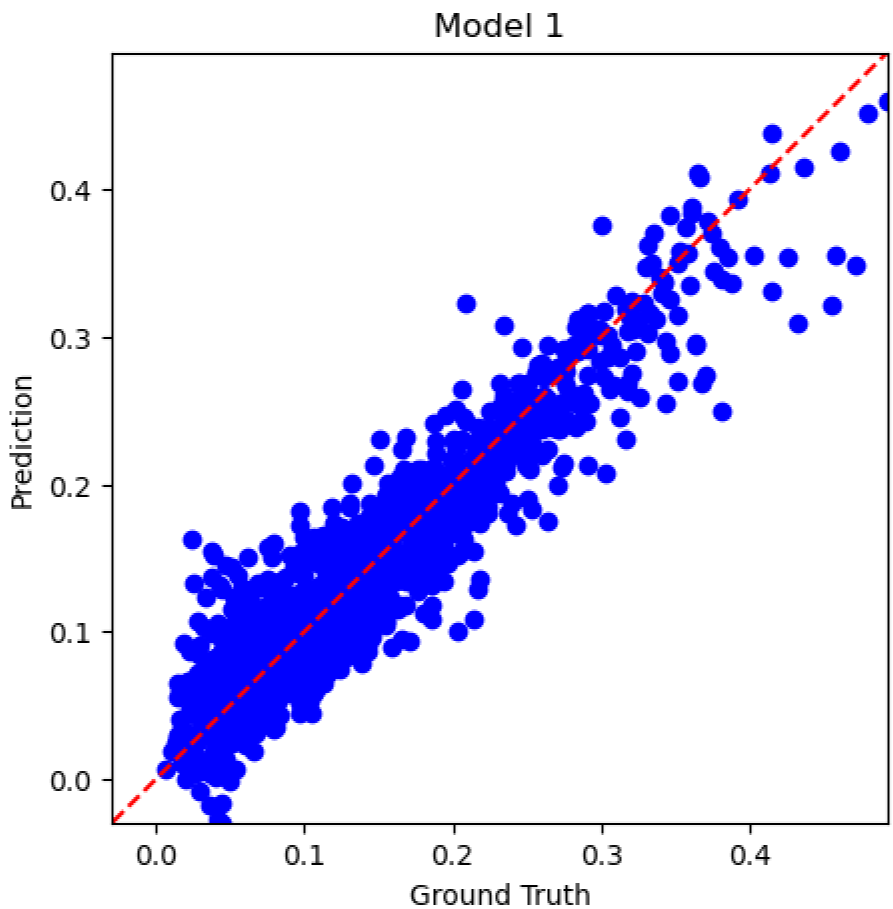
```
R_squared for Model 1: 0.872799404155802
R_squared for Model 2: 5.806477521019815e-09
R_squared for Model 3: -2.8738726437506803
```

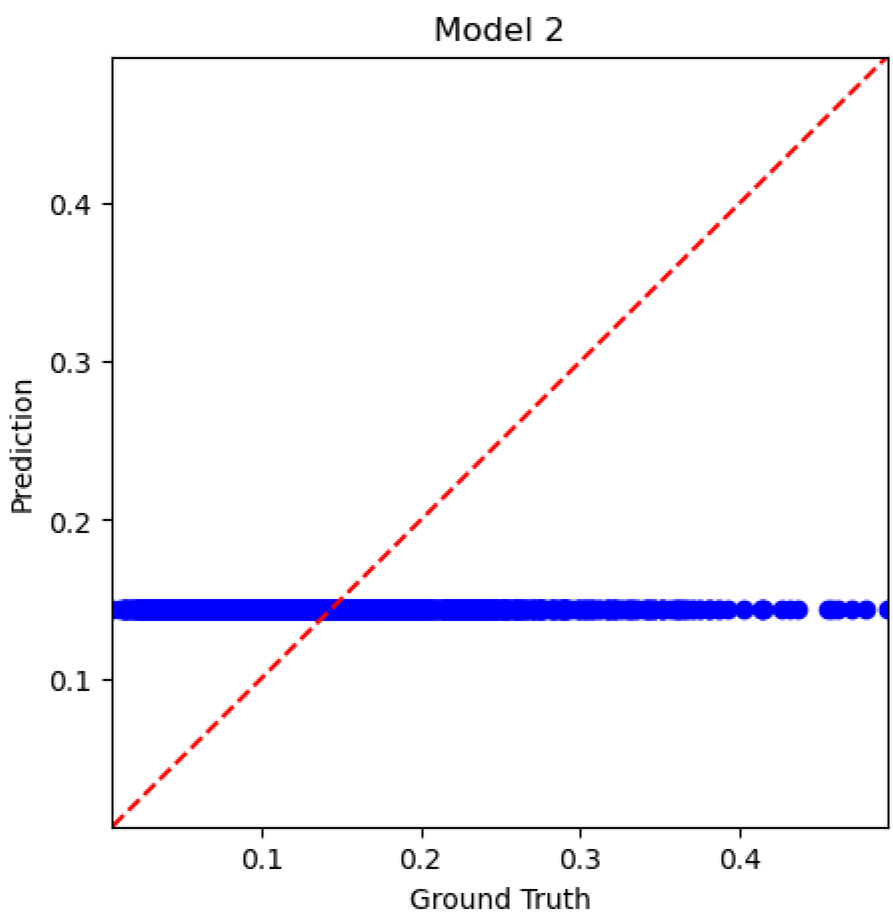
Plotting predictions vs ground truth

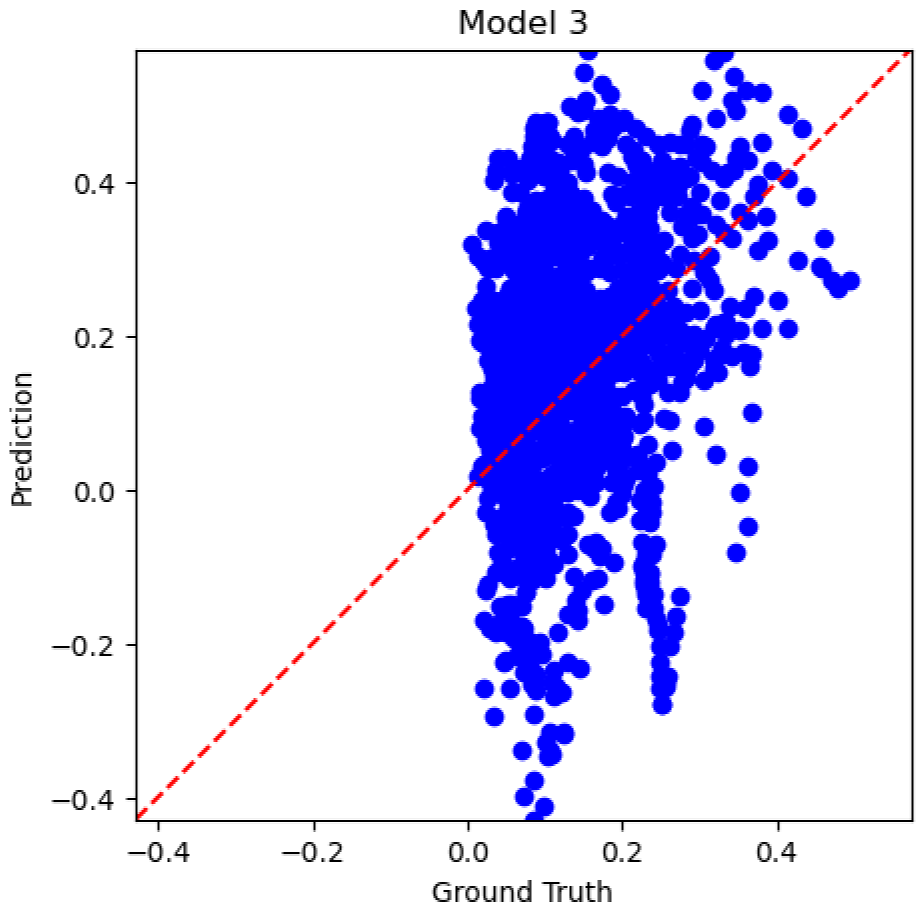
Complete the function definition below for `plot_r2(gt, pred, title)`

Then create plots for all 3 models.

In [7]:
```
def plot_r2(gt, pred, title):
    plt.figure(figsize=[5,5])

    # YOUR CODE GOES HERE
    plt.scatter(gt,pred,c='blue')

    plt.plot([-1000,1000], [-1000,1000],"r--")

    all = np.concatenate([gt, pred])
    plt.xlim(np.min(all), np.max(all))
    plt.ylim(np.min(all), np.max(all))
    plt.xlabel("Ground Truth")
    plt.ylabel("Prediction")
    plt.title(title)
    plt.show()

plot_r2(gt, model1,"Model 1")
plot_r2(gt, model2,"Model 2")
plot_r2(gt, model3,"Model 3")
```


Model 1
Prediction
Ground Truth
0.0
0.1
0.2
0.3
0.4


Model 2
Prediction
Ground Truth
0.1
0.2
0.3
0.4



Questions

1. Model 2 has an R^2 of exactly 0. Why?

The formula of R^2 is 1- SSR/SST where SSR is sum of squared residentials and SST is total sum of squares. R2 is 0 which means that SSR = SST. This means that model is not capturing any relationship between the dependent and independent variables. Looking at the dataset as the values of model are all same so the SSR gives the same value as the mean of all the model subtracted from ground truth which is the same as SST. Another reason is that the sample size is small which can lead to a bad R^2 value. Along with that, the model would have been trained poorly which led to the constant values in the predicton therefore giving us a value 0. Furthermore, the model may be built using only one feature which can also lead to constant predictions therefore SSR being equal to SST. Hence, the value of R^2 is 0.

1. Model 3 has an R^2 less than 0. What does this mean?

Model 3 has a R^2 value less than 0 which can occur because the model fits the data worse than simply reporting the mean of ground truth as the model's output. If the error is large then the SSR value is large which gives R^2 less than 0. As the prediction values varies a lot therefore the mean of the prediction value is high which leads to a higher SSR and hence the R^2 value is less than 0. A poorly chosen model can lead to a value less than 0.

In [ ]:

```
```